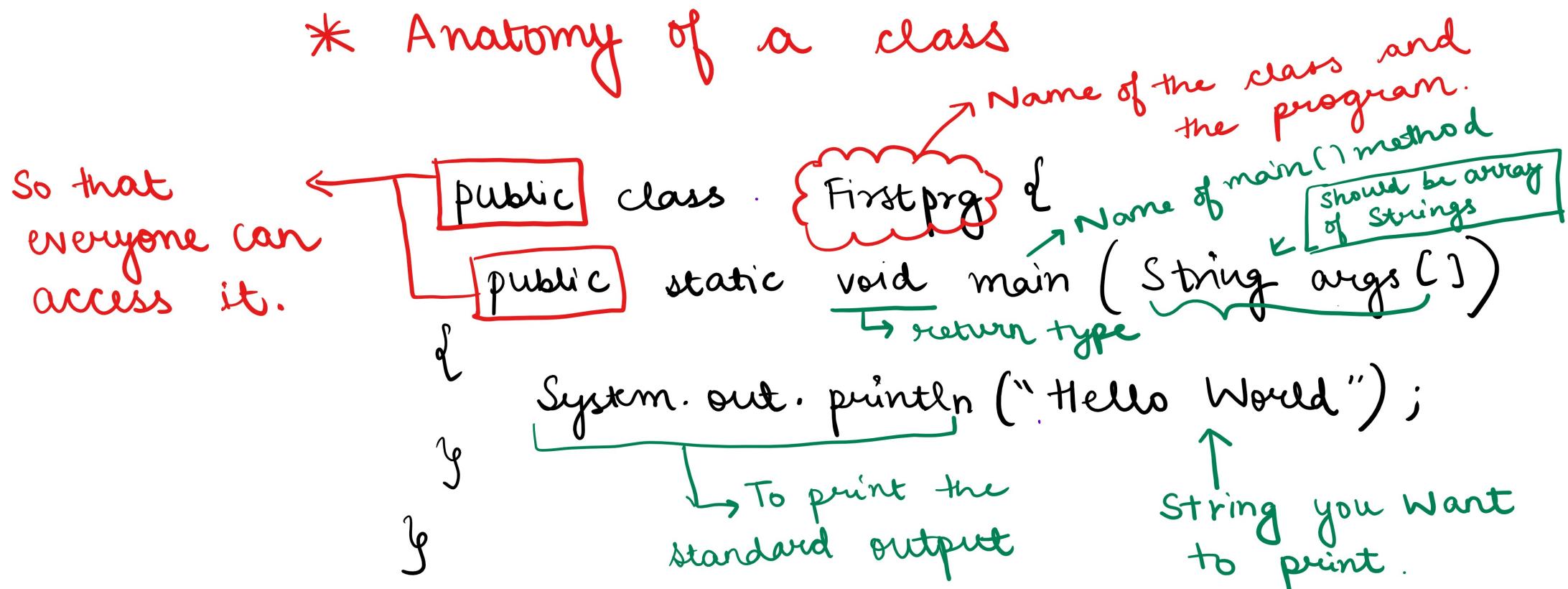


Ques. *① [. In Java, everything goes inside a class. . .
When we are running a program, we are simply running a class!]

② No matter how long our code is, there should be a main() method so that our program would run.



Object Oriented Programming (OOPS 1)

① Introduction to OOPs.

→ Object Oriented Programming System (OOPS) is a paradigm which is based on the concept of "classes" and "objects" that contains data and methods.

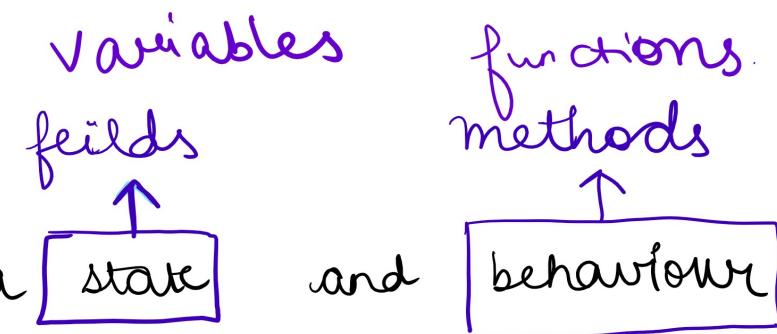
It is used to structure a software program into simple, easy to reuse piece of code called "CLASSES".

→ What is an Object ?

The object is an entity which has a state and associated with it.

They are based on real-world objects.

The Object is an instance of a class.



⇒ For example, let's consider a class 'Vehicle'

```
class Vehicle {
```

```
    int gear = 5;
```

```
    public void brake()
```

```
    { System.out.println("Stop the vehicle"); }
```

```
}
```

```
}
```

This class can have objects like car, bus, motorbikes, and bicycles.

Subclass

Car ()
Motorbike ()

We can create an object like :

```
Vehicle
```

obj

```
= new
```

```
Vehicle();
```

constructor of the class.

Name of the Class

Name of the object

new keyword is used to allocate memory

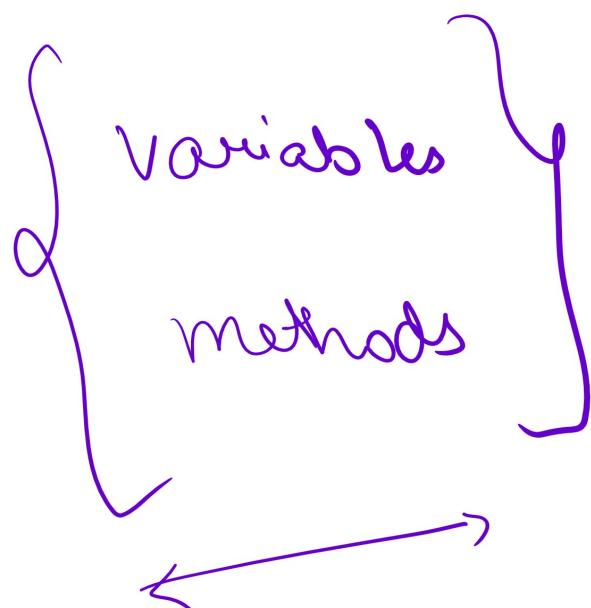
Object Oriented Programming Language.

- It is a methodology used to design a program using **CLASSES** and **OBJECTS**

WHAT IS A CLASS?

- class is a blueprint of how our object would look like. The blueprint defines the variable and methods.
properties behaviour.

① Encapsulation : It is defined as the wrapping up of data under a single unit.



```
func1()  
func2()  
Void main()  
{  
    int temp;  
    int node;  
}
```

individual unit

②

Inheritance: It is a method by which

one class can inherit the properties
of another class.

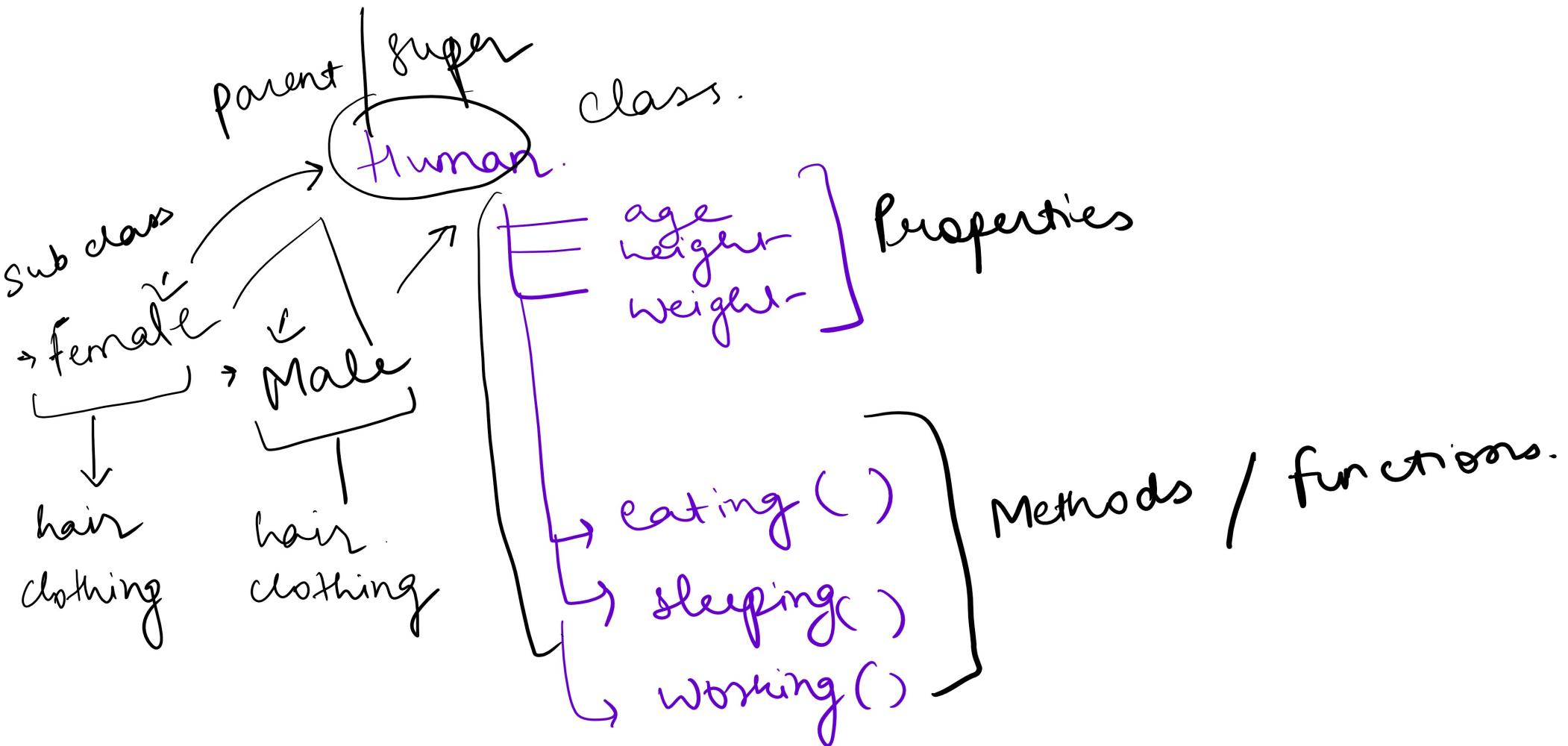
Sub class → child

the class which
inherits the property

Super class → parent.
methods variables

whose properties are being
inherited.

IS A Relationship



⇒ Syntax for Inheritance in JAVA

```
class parent ()  
{  
    //  
}  
  
class child extends parent ()  
{  
    //  
}
```

